5224ch01.qxd_jd 6/28/01 11:22 AM Page 1

FOR PUBLIC RELEASE

ONE

Samba and Windows

Windows® network administrators have heard about Samba®. It's the free software that makes a Linux® or Unix® computer look like a Windows server on a network. As a network administrator, maybe you've decided to try it out and quietly slip a Linux server running Samba onto the local network. That's already been done with thousands of Linux servers. On the other hand, setting up a Samba server may not be your idea. Maybe your boss just asked why you aren't using one of those free Linux servers everyone is talking about.

Whatever the reason, the time has come to set up and run a Samba server.

Increasingly, system administrators are seeking out open source software like Samba and Linux because it is hassle-free. There's no complicated licensing to maintain and no additional paperwork involved when adding users or upgrading systems. For a busy administrator this is no small advantage.

Also, Samba can be installed on an older computer that would be too slow for a Windows 2000 server, but would give excellent performance running Linux. This kind of savings on hardware costs is another reason for Samba's popularity.

In this chapter we'll go over some of the advantages of free software like Samba and Linux and the reasons for Samba's legendary stability. Also, we'll compare Samba and Windows, seeing what is similar and what is different. What's different is important because there are some things that a Samba server just can't do. If you only skim through this chapter, make sure you read this part and make sure that there isn't a limitation that prevents you from using Samba.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 2

The Secret to Free High-Quality Software

Samba, like the Linux operating system, is licensed under the GNU General Public License, often shortened to GPL. This is a software license established in 1984 by the Free Software Foundation. A copy of the license can be found in Appendix D

The license basically says that Samba can be freely distributed by anyone without restriction. The code is freely available to read, change, fix, give away, or sell as long as nothing is done that would restrict any of these rights. However, Andrew Tridgell, the original developer of Samba, and others who have contributed to its development retain the copyright to the Samba code.

Few network administrators will ever look at the Samba source code. So you might wonder why you should care whether the code is openly available or not.

While you may not ever look at the code, there are those who can and will look at the code, particularly if they are having a problem and they know enough to be able to fix it. Once a fix has been written, it can be made available to everyone else who is using the software. There is nothing new in this. IBM's most stable systems were originally built this way, and most computer software was developed the same way until fairly recently. Bug fixes and enhancements were freely exchanged, making possible the rapid development of very stable software.

Without shared code, software development would have taken much, much longer. The Free Software Foundation's open source licensing is designed to encourage this kind of software development so that progress isn't throttled.

The integrity and stability of open source software are enhanced by the peer review process, the same process that underlies the reliability of all scientific research. Open source software must stand up to the most demanding public scrutiny. All code is available for anyone to examine, verify, and use. Buggy code is either fixed or discarded. Innovations are added based on the suggestions of users. This kind of testing can be more extensive than is possible in the closed environment of proprietary software development.

In fact, closed software development often hinders the process of debugging software. Testing is often limited so that confidential developments are not revealed.

The open source development model makes it almost impossible to hide defects in software, and open source software will never be able to secretly collect information from users' computer systems. Open source prevents a software developer with a vested interest in the product from also being responsible for testing the software's capabilities and then verifying its usability.

The Internet itself is proof that open source works. The Internet is built on open source applications from the TCP/IP networking protocol—used by every Microsoft Windows computer on a network—to the Apache Web server software, used on more than half the Web servers in the world.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 3

Samba Put Free Software onto Office Networks

Samba and Linux have grown up together. The quality and reliability of Linux made free open source software a viable alternative in computer operating systems. An operating system, however, is only as useful as the software applications available to run on it. Would Microsoft's operating system have been as successful without the Microsoft Office suite of applications? Bill Gates certainly doesn't think so.

Linux owes its success, in part, to two free open source applications: Samba and the Apache Web server. Samba's role was often hidden in the early use of Linux servers. Samba was quietly being installed by system administrators on Intel-based computers running Linux. To the users on the network, these servers looked just like Windows servers.

Word quietly spread that older computers could be set up with Samba and become a seamless part of a Windows network. This was an exciting development, especially for cash-strapped institutions like universities and public schools. Educational institutions became the proving grounds for Samba servers. These servers were put to the test. In many respects, the demands of a university computer network can be more exacting than those of a corporate network. Academic uses can be more varied, and students tend to be more abusive of the system than business users. Samba met this test.

Samba file and print servers have won the endorsement of computing giants like IBM and Hewlett-Packard. Samba servers have also won over the bean counters. That's because Samba makes management look smart. Managers can choose the top-rated technology and still protect the bottom line. Samba servers eliminate the need to spend money on client licenses. Running Windows 2000 Advanced Server with 100 client licenses costs about $9,400. In contrast, a Linux server running Samba with unlimited users costs about $50.

Why Is It Called Samba?

Commercial software companies can spend a bundle of money audience-testing names for a new product. They want just the right name that will enhance sales.

The names used for free software, on the other hand, are not market-tested. Free software often follows the hacker tradition creating a play on words or a recursive acronym. The Free Software Foundation's GNU software license uses such a recursive acronym. GNU stands for GNU's Not Unix.

Samba is not a recursive acronym, and the name was never market-tested. Samba was originally called SMBserver, a utilitarian name. SMB is the Windows networking protocol. Microsoft has enhanced the original SMB protocol and calls the enhanced version CIFS (Common Internet File System), but it is still the SMB protocol. So SMBserver was a logical choice for the name. However, it turned out that the name had already been trademarked

5224ch01.qxd_jd 6/28/01 11:22 AM Page 4

by a company that makes a competing product. When Andrew Tridgell got an email message telling him that he had to stop using the name, he needed to find a new name quickly. With no idea what to call the program, Tridgell ran a search through the dictionary on his computer for every word that started with the letter "s" and included the letters "m" and "b."

From the list of words that he got, Tridgell chose Samba. The name may not have been market tested, but it works at least as well as the ambiguous Windows NT or the even stranger Millennium Edition (Windows Me).

The History of Samba

Samba was developed beginning in 1991 as a solution to a problem Tridgell had while he was a graduate student at the Australian National University. He had a PC, but he needed to access some disk space on a Unix server using the Pathworks protocol from Digital Equipment Corporation (DEC). The problem was, there was no Pathworks client for his DOS-based PC. Being a skilled programmer, Tridgell studied the problem and then hacked together some code to do the job. With his code he could access the Unix server as if it were another PC DOS computer.

That was version 0.1 of Samba. The Pathworks protocol, it turned out, is the same SMB protocol used for Windows networks.

Tridgell has written a short account of the beginnings of Samba, which can be found in the document named *history* included with the standard installation of Samba. The document can be read on the Web at *http://us1.samba.org/samba/ftp/docs/history*.

Tridgell released the code over the Internet in 1992. A few months and a few fixes later, version 1.0 was released. The code remained unchanged for the next two years. Then, Tridgell discovered Linux. He wanted to link his wife's Windows PC to his Linux PC. There was no easy way to do this without purchasing additional software for the Windows PC. He decided to try the early Samba code and was amazed to find that it worked.

At about this same time, the version 1.0 code was being discovered by Linux users. Tridgell started getting email from Linux users around the world who wanted to use the Samba code to connect Linux and Windows computers. Soon volunteer programmers were offering to help develop the code. There was a spurt of development, the major bugs were worked out, and within two years Samba running on Linux matched the performance of any of the commercial server products that were available for networking Unix systems with Windows.

Today, Samba is found on computer networks around the world, from simple home networks to sophisticated corporate networks. Samba has been accepted by almost every major Unix vendor as a standard for Unix-Windows connectivity. Hewlett-Packard (HP) has gone even further.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 5

HP is distributing Samba with its HP-UX operating system. HP's Samba distribution is called CIFS/9000. According to a statement from Hewlett-Packard, not only is "CIFS/9000 based on open source Samba, [but] HP is committed to submitting CIFS/9000 enhancements back to the open source community." This has already happened, and Samba now includes enhancements developed by HP's programmers. This is an important endorsement of both Samba and free open source software.

Similarly, IBM has embraced Samba and Linux. The company has already earmarked over $1 billion for Linux development, and it is developing an open source version of its JFS filesystem for Linux. All of IBM's Linux servers use Samba for connecting with Windows systems.

As for Microsoft, it may not officially endorse Samba, but several of its developers helped make Samba a success. A Samba document named *thanks* expresses Andrew Tridgell's appreciation to people who did not write code or make bug reports, but who made significant contributions to the development of Samba. His thanks begin with a list of five Microsoft employees who were "very helpful and supportive of the development of Samba" over many years. The document can be read on the Web at *http://us1.samba.org/samba/ftp/docs/THANKS*.

Comparing Samba and Windows

Samba is designed to be run on a Linux computer or on almost any of the Unix systems. It is a Unix implementation of the Windows networking protocol. This blend introduces differences that can be desirable for your network as well as some that you may not want. Running a Linux server can put additional demands on network administration and may add costs for training and implementation. On the other hand, the overall cost of adding a Linux server running Samba to your network, including training, may be lower than the cost of adding a Windows 2000 server. After all, using the example cited earlier, a $3,000 training course in Linux system administration is a third of the cost of $9,400 for the Windows 2000 server software and client licenses for 100 users. In addition, a Samba server has some strengths that aren't available on Windows systems.

Here's a breakdown of some things to consider when deciding whether Samba is the right choice for your network.

Samba's Strengths

Samba has a lot going for it. It is a stable, proven platform for providing file and print services on a local area network. The cost for a Samba server is low and administration is simplified. Adding a Samba server will also increase the security of your network.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 6

LOWER COSTS FOR SOFTWARE AND HARDWARE

There is no base cost for Samba. It is included with every Linux distribution. Prices start at about $50 for a CD with the operating system and all the additional software you'll need, plus a basic user's manual. If you purchase a computer system with Linux already installed, Samba is usually included. If not, all the Samba software can be downloaded for free over the Internet from the Samba Headquarters at *http://www.samba.org* or from any site that has Linux software.

A Linux server has lower hardware requirements than a Windows server. Linux can even be run on older systems that will not support Windows 2000. Because Linux does not require a graphical interface and can be run in terminal mode, an old Pentium 100 system with 32 megabytes (Mb) of random-access memory (RAM) can make an excellent file and print server for a small workgroup that needs its own server. The graphical interface is a resource hog that uses not only video memory but also system memory and virtual memory. Even on a top-of-the-line server, Linux can exceed the performance of a Windows server. Some controversial tests commissioned by Microsoft concluded that on single-processor systems, a Linux server running Samba will outperform a Windows NT 4 server. The same tests showed that Windows servers outperform Linux on multiprocessor systems. The Linux 2.4 kernel that was released in January 2001 has significantly improved multiprocessor support.

Of course, software and hardware are not the only costs involved in maintaining a network, and you wouldn't choose Samba over Windows just on that basis. But Samba is lower in cost even when you consider technical support agreements, the price of upgrades and service packs, downtime costs, and personnel costs for system administrators.

OPEN SOURCE LICENSING

The GNU public license is extraordinarily simple to comply with. No paperwork is involved, and there are no records to keep. It doesn't matter if your Samba server has one user or 10,000; no per-user or per-workstation licenses are involved.

This is good news for network administrators. There has been an increased burden on administrators as software licensing for most operating systems and applications has become more complex.

In order to make sure that there are no violations of software license terms, administrators are too often tied down with the paperwork involved with manual tracking of licenses. Most administrators work to keep their systems within the law, but that can be a time-consuming task, and time is the one thing most administrators don't have. The alternative is expensive license-tracking software.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 7

VENDOR INDEPENDENCE

With an all-Windows network, you are locked into total reliance on Microsoft. That's not all bad. Microsoft has set standards, and standards are welcomed by most network administrators. The software standards greatly reduce the learning curve for any new software, and the hardware standards improve reliability and compatibility.

Yet there are problems with this reliance on a single company. For example, when something isn't working right with an application, users have to pay a service fee to Microsoft even if it is just to learn that there is a known problem with the software. The Justice Department, in its case against Microsoft, has shown some of the downside of Microsoft's total domination. Left out of the legal dispute are Microsoft's plan to start charging yearly fees for software use as well as the company's monopoly pricing practices.

Using the combination of Samba and Linux gives a network administrator independence from a single software monopoly without sacrificing standards. Linux follows the same Unix-based standards that the Internet is built on and that Microsoft has been adopting to a great degree. Support is available from a number of vendors. For example, both IBM and Hewlett-Packard offer 24/7 service support for Linux and Samba.

IMPROVED NETWORK SECURITY

Security is a big concern for all network administrators. No system is invulnerable, including both Windows and Linux. Having a single operating system throughout your network makes you much more vulnerable.

Just as biological diversity in the environment prevents the transmission of viruses between species, systems diversity blocks the spread of computer viruses and the like between systems. Most attacks on systems rely on vulnerabilities of a single operating system. Mixing up the operating systems thwarts some of the most common forms of attack.

In fact, Microsoft uses systems diversity to protect its software. In the document *How Microsoft Ensures Virus-Free Software,* which can be read at *http://support.microsoft.com/support/kb/articles/Q80/5/20.ASP,* Microsoft says: "Disks are duplicated on a variety of industrial strength, quality focused systems. Most of these systems are Unix-based. The Unix-based duplication systems used in manufacturing are impervious to MS-DOS-based, Windows-based, and Macintosh-based viruses."

Take a tip from Microsoft and diversify. Mix some Samba servers with your Windows servers and give your network greater security.

WEB-BASED ADMINISTRATION

Remote administration is an expensive add-on for a Windows NT server. Windows 2000 includes a telnet server that allows remote access, but command-line administration of a Windows 2000 server isn't easy or pretty.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 8

A Samba server can be securely administered through Web-based applications that are included with many Linux distributions or available for free over the Internet. Samba system administration is possibly even easier than administering a Windows NT server. You can sit at one terminal and administer one or a hundred Samba servers all from the same workstation. You can even do Samba and Linux management securely from home over a simple 56K dialup connection to the Internet.

Two applications—the Samba Web Administration Tool (SWAT) and Webmin—will give you everything you need in the way of system administration tools. You'll find all the details on how to use them in this book.

The only way you can do the same job with Windows servers is to purchase expensive third-party software, most of it requiring a high-bandwidth connection to use.

LINUX'S LEGENDARY STABILITY

Overall system performance is mostly determined by hardware, but stability and reliability are based on the operating system. The Linux operating system is legendary for its stability and reliability.

Linux inherits its stability and reliability from its Unix roots. Because of the Unix architecture, Linux rarely crashes, becomes unstable, or loses or corrupts data. The Unix architecture is completely modular, that is, its software components are clearly separated, with well-defined interfaces between components. The modular architecture not only makes it easier to debug code but makes it much harder for a bug in one component to affect other components or to crash the whole system. With this design, a single point of failure does not translate into a failure of the whole system.

The Linux system kernel is completely independent of the user interface. This independence reduces the likelihood of needing to reboot the system when there is a problem with the video interface. The Linux graphical interface, called Xwindows, is a separate server module that can be independently rebooted if necessary without rebooting the operating system itself.

In addition, the Linux architecture makes it easy to eliminate unnecessary overhead. If you don't need the graphical user interface (GUI), which is unnecessary on a Samba server, you don't have to run it, and RAM and disk space that would normally be required to support the interface can be made available for running what you do need, like Samba services.

Where Samba and Windows conflict

This is not a Samba versus Windows comparison. Just as there are reasons to choose Samba, there are reasons why it may not be the right choice for you. Open source software is successful because it is based on the principle that one size does not fit all. Free software is about freedom of choice. There are

5224ch01.qxd_jd 6/28/01 11:22 AM Page 9

many reasons to choose Microsoft, even if "there's no other choice" shouldn't be one of them.

Microsoft sets the standards for Windows servers, and Samba is always behind in adding any new features. If you need to have the latest features from Microsoft as soon as possible, then choose a Windows server. Also, consider the additional skills required to support both Windows and Samba before choosing to add a Samba server to your network.

MICROSOFT SETS ITS OWN STANDARDS

Microsoft regularly updates its products, including incremental service packs for both server systems and applications. These updates often introduce new features. Sometimes the new features might be exactly what you need for your network.

Samba can only play catch-up because Microsoft sets its own standards and defines the new features to be added to a Windows network. There is always a lag time between anything new Microsoft has introduced and the availability of that feature in Samba. It is even possible that some feature may not be added to Samba. If you must always have the latest features that Microsoft offers, then Samba is not be the right choice for you.

SAMBA IS LIKE NT 4, NOT WINDOWS 2000

Think of Samba as a Windows NT 4 file and print server. Some of the features found on a Windows 2000 server are not supported by Samba, such as the Active Directory. Active Directory is based on Lightweight Directory Access Protocol (LDAP) but uses its own access control system and replication methods. Like an NT 4 client or server, Samba cannot participate in Active Directory user management, and the Active Directory tools on a Windows 2000 server can't be used to administer a Samba server. This will change with the development of Linux LDAP and Windows Active Directory interoperability, which is being done by another open source team, the OpenLDAP project (see *http://www.OpenLDAP.org*).

Another feature on a Windows 2000 Server that isn't directly supported by Samba is the Kerberos security key system for user authentication and authorization. Although the Kerberos system originated on Unix systems, the Microsoft implementation is not compatible with Linux or Unix systems. You can't use a Windows 2000 server to authenticate Linux users or servers, and you can't use a Linux Kerberos server to authenticate Windows 2000 servers or users. Work is also being done to develop a version that will make a Linux Kerberos server compatible with Windows 2000.

WINDOWS SERVERS ARE BEST FOR DOMAIN CONTROLLERS

A Samba server is seen as an NT 4 server by a Windows 2000 server. The Samba server cannot be a domain controller on a network that includes Win-

5224ch01.qxd_jd 6/28/01 11:22 AM Page 10

dows 2000 domain controllers. If you already have a Windows domain controller, there is no reason to switch to a Samba domain controller. You should switch to a Samba domain controller only if you are removing all of your Windows servers and replacing them with some other operating system, such as Linux. In general, if you need a Windows domain controller you should use a Windows server.

WINDOWS HAS FINER ACCESS CONTROL

There is no way to directly translate Windows security IDs and access control lists (ACLs) to the user account and file permissions system that Linux uses.

The Windows NTFS filesystem supports much finer control over who can do what than is possible on a standard Linux system. Windows ACLs treat users as individuals who can be given rights to files and directories in any combination. Groups can contain individuals or other groups. Windows has six access rights that can be defined for the user, for a group, or for everyone. These rights are Read, Write, Execute, Delete, Take Ownership, and Change Permission. For example, a file can be set to be viewed by two users in a group and both viewed and modified by a third user in the group, while a fourth user is denied all access to the file.

The Linux *ext2* filesystem does not include ACL support. The Linux filesystem defines Read, Write, and Execute permissions for the user, the group, and others. A number of options are available to add ACL support on Linux systems, but there is no standard implementation. No Linux distribution includes this feature in its standard kernel. On systems with ACL support compiled into the kernel, Samba will recognize the extended ACL rights. This setup gives permission rights similar to those used on Windows servers, but the ACLs are not directly translatable from one system to the other.

SAMBA REQUIRES ADDITIONAL SKILLS

If you or someone on your staff doesn't already know how to support Linux, then adding a Samba server means there is a new system to learn and support. The most difficult part of supporting a Samba server is the initial installation of Linux and the configuration of Samba. Once the server is set up, it is as easy to maintain as a Windows server, including basic troubleshooting. Dealing with any complex problems that might arise, however, can be difficult without Linux or Unix experience.

WINDOWS HAS MORE HARDWARE SUPPORT

There is a vast array of hardware choices for Intel-based PCs. All vendors make sure to include any necessary software drivers for Windows systems with hardware. Linux drivers, on the other hand, are not always available. If you have special hardware needs, check to make sure that it is supported by

5224ch01.qxd_jd 6/28/01 11:22 AM Page 11

Linux if you need to have it on your Samba server. For example, there are no Linux drivers for digital video disk (DVD) drives because of legal prohibitions. If you need to have a shareable DVD drive on your file server, this can only be done from a Windows system.

Linux and Windows

The differences between Linux and Windows are often exaggerated. Both are operating systems working on the same computer hardware. Both must carry out essentially the same or similar tasks.

To successfully integrate a Samba server into a Windows network requires some familiarity with the Linux operating system. You'll be surprised at how many Windows administration skills work in the same or a similar way on Linux. Many Linux terms and concepts have Windows equivalents. For example, the ominous-sounding Linux daemons are similar to services on a Windows server. In fact, system administration isn't really operating system–specific.

Following are descriptions of some Linux features and terms, with explanations. Where appropriate, Windows equivalents are given.

LINUX IS CASE-SENSITIVE

The Linux operating system is case-sensitive: for example, *letter.txt* is different from *letter.TXT*. Windows NT/2000 recognizes case and preserves it but is case-insensitive. Using Windows, you could request a file named *LETTER.TXT* and get *letter.txt* and *letter.TXT*, but using Linux you'd get a "File Not Found" error. Case sensitivity is one of the biggest differences between the two operating systems when it comes to file naming.

LINUX USES THE FORWARD SLASH

On a Linux system you don't use the backslash as a file seperator, you use the forward slash. Windows is built on DOS, which borrowed many features from the old CP/M operating system that used the backslash as its file separator. Linux is written in the C language, which uses the backslash for another purpose and uses the forward slash for its file separator. For example, at a command prompt in Windows you might enter `cd \personal\letters` and on Linux you would enter `cd /personal/letters`. In Linux (and the C language), the backslash indicates that a character is to be used literally. Thus, entering the same command on Linux using a backslash, `cd \personal\letters` would be interpreted to mean `cd personalletters`. The forward slash is also the default file separator on all Web pages because the Internet was initially built on Unix systems.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 12

LINUX HAS NO UNERASE FEATURE

On a Linux system, if you delete a file it is gone with no possibility of retrieving it. After a file is deleted its free space is immediately made available for something else. This improves multiuser performance but sacrifices the convenience of retrieving deleted files. Users of Windows systems won't expect this behavior.

LINUX DOESN'T REQUIRE FILE EXTENSIONS

File extensions are optional on Linux and are used mainly for ease of identification. A file extension on Linux does not affect whether the file is executable or any other possible aspect of a file. This fact also means that file types aren't associated with extensions. The window managers in Linux, such as the K Desktop Environment (KDE) and Gnome let you define MIME types that associate file extensions so that clicking on a file with a certain extension will automatically open the associated application. This function is not, however, built into the Linux filesystem and has to be individually defined. As with Windows, file associations in KDE and Gnome don't always work correctly, and there can be conflicts when more than one application uses the same extension.

LINUX DOESN'T USE DRIVE LETTERS

Windows uses drive letters to indicate disk drives, partitions, and shares on servers or other Windows workstations. Samba lets you map a share on a Linux server and assign it to a drive letter on a Windows machine. But Linux does not use drive letters at all. On a Linux system everything appears to stem off a single directory hierarchy. Rather than mapping to a drive letter, disk drives, partitions, and shares on other servers are mapped to a mount point that is part of the overall directory structure. (The terms "mounting" and "mount point" used on Linux and Unix systems come from older systems that used tape reels that had to be mounted before files could be accessed.)

Devices such as hard drives, network cards, and video cards also show up on Linux as part of the directory structure. These devices all look like files in the directory named *dev*. For example, the first primary disk drive on an ATA controller is a file named *hda* in the *dev* directory. This is usually written out as */dev/hda;* the second disk drive is named */dev/hdb*. On Windows, these might be the C and D drives. If you divide the first drive into two partitions, on Windows you might then have C and D on the first drive and E on the second drive. On Linux these would be */dev/hda1,* for the first partition of the first drive, and */dev/hda2* for the second partition. The second drive, if partitioned as a single drive, is named */dev/hdb1*. Table 1-1 lists the Linux

5224ch01.qxd_jd 6/28/01 11:22 AM Page 13

Table 1-1 *Common device names in Linux*

Device	Linux name
Floppy disk	*/dev/fd0* or */dev/floppy*
First drive on primary AT Attachment (ATA) interface	*/dev/hda*
First partition on first drive on primary ATA interface	*/dev/hda1*
Second partition on first drive on primary ATA interface	*/dev/hda2*
Second drive on primary ATA interface	*/dev/hdb*
First drive on secondary ATA interface	*/dev/hdc*
Second drive on secondaryATA interface	*/dev/hdd*
First Small Computer System Interface (SCSI) drive	*/dev/sda*
Second SCSI drive	*/dev/sdb*
SCSI tape drive (rewinding)	*/dev/st0*
SCSI tape drive (nonrewinding)	*/dev/nst0*
CD-ROM	*/dev/cdrom* or the ATA/SCSI name such as */dev/hdb* or */dev/sda*

names for several common devices. The device names are standard on the Linux 2.2 kernel. These names are also supported by the 2.4 kernel, although a new naming convention is being adopted to allow for enhanced support of USB devices.

SOME COMMON LINUX QUESTIONS

HOW DO YOU GET ADMINISTRATOR ACCESS? • The administrator logon name for Linux is "root." Unlike Windows, there is no equivalent of the Administrators group that gives administrator privileges to individual users. Linux has only one administrator account, which means that multiple users with administrative responsibilities usually must know the administrator's password.

WHAT ARE LINUX DAEMONS? • Linux daemons, which function similarly to Windows services, are services that are activated at system startup and run

5224ch01.qxd_jd 6/28/01 11:22 AM Page 14

until the system is shut down or rebooted. The name comes from Greek mythology: daemons were guardian spirits, and Linux daemons work in the background as service guardians. Each daemon has its own configuration file or command-line options, or both. As with Windows, each service has its own daemon. The typical daemons started by Linux include a print spooler, a mail handler, system loggers, a task scheduler, and various network services. Samba is typically run as a daemon on a Linux server.

WHERE ARE THE LINUX STARTUP FILES? • Most Linux systems run an initialization script at startup to start the daemons. There are usually several different scripts, one for each runlevel. The six runlevels used by most Linux distributions can be compared to the levels shown on Windows startup when you press the F8 key. Windows runlevels include DOS, Safe Mode, Safe Mode with Networking, and so on. Linux runlevel 1 is like DOS; no daemons are started. This is a level used only for system maintenance. Runlevel 2 is like Safe Mode; some but not all daemons are started. Level 3 might be likened to Safe Mode with Networking, though it is much more than that. This is the full system running and is the mode used for many servers. Not only is networking started, but all of the daemons as well. Finally, runlevel 5 is the full Windows mode. The only difference between levels 3 and 5 is the GUI that is automatically started at runlevel 5.

WHERE IS THE LINUX TASK MANAGER? • Linux does not have a single tool that performs all the tasks of the Windows Task Manager. Linux has individual tools for every task. These tools are all accessible through a single interface, the Webmin software. Webmin is a Web-based administration interface for Linux. It can be used to start or stop a service, to monitor service status and resource usage, or to alter the priority of a service.

WHERE IS THE LINUX USER MANAGER? • As with the task manager, Linux does not have a single tool that is the user manager. However, all the tools for creating and maintaining user and group accounts as well as permissions can be accessed through the Webmin system administration software.

WHERE'S THE REGISTRY? • Linux does not have a system registry. Some of what is found in the Windows registry can be accessed on the Linux */proc* filesystem. This is not a true filesystem, but rather an interface into the kernel and the currently running processes. As in the Windows registry, changes can be made to the */proc* filesystem, but the changes are effective only on the currently running system. When the system is shut down or rebooted, the changes are lost. To make a change to */proc* occur on each startup, the administrator must include it in the initialization scripts. Some registry-type settings can also be made through the system configuration files found in the */etc* directory. Many of these configuration changes can be made through Webmin; some changes can only be made by using a text editor and putting

5224ch01.qxd_jd 6/28/01 11:22 AM Page 15

them into the configuration file manually. Windows also stores passwords and group information in the registry, which are found in the */etc/passwd, /etc/shadow,* and */etc/group* configuration files on Linux.

WHERE IS THE EVENT VIEWER? • The Linux Syslog daemon starts at boot time and keeps logs of all system messages. The log records are similar to the system events shown by the Windows Event Viewer. All the logs are saved in the */var/log* directory and can be viewed through Webmin or by using a text viewer.

HOW DO YOU DEFRAGMENT LINUX DRIVES? • Windows administrators, who are used to dealing with fragmentation issues on their servers, will be glad to know that fragmentation is kept low on Linux hard drives and defragmentation is seldom necessary. When Windows NTFS writes a file, it can be broken up into multiple clusters and spread across the hard drive. This approach can speed disk writes because writing can begin with the first available space. However, it also fragments the file contents and can make access inefficient. Defragmentation, the process of putting the pieces of all the files together so that each file is located on a single continuous series of clusters, can often significantly improve performance.

The Linux *ext2* filesystem uses a different approach. It keeps a list of all empty blocks on the hard drive. When blocks are needed to write a file, *ext2* looks at the list and finds a contiguous set of free blocks that can hold the entire file, if one is available.

Over time, however, all filesystems tend to become fragmented. Full drives and busy network drives can become fragmented on a Linux system. The usual method for defragmenting a Linux hard drive is to do a complete backup, reformat the partition, and then restore the files from backup. When the files are restored, they are written in contiguous blocks.

HOW DO YOU CREATE SHORTCUTS IN LINUX? • Shortcuts in Linux are called links. Links can be created from the command prompt or by using one of the file managers. The command is `ln`. For example, to create a simple link from a database in the finances directory to a directory named projects, enter this command:

```
ln -s /finances/expenses.db /projects/expenses.db
```

This command creates a shortcut in the projects directory called *expenses.db*. The *-s* flag indicates that it is a symbolic link, that is, a link that points to the location of the actual file. It works in many ways like a Windows shortcut, with some exceptions. One of the limitations of links is the restrictions imposed by permissions. In order to access a linked file you must have permission not in the directory where the link is found, but in the directory where the actual file is located.

5224ch01.qxd_jd 6/28/01 11:22 AM Page 16

WHERE ARE THE HIDDEN FILES IN LINUX? • The Linux filesystem doesn't have a hidden file attribute. Instead, filenames that begin with a dot, such as *.profile*, are interpreted as hidden by most applications when displaying a list of files. This is similar to the Windows Explorer setting of not displaying any files with system file extensions, such as *.dll* or *.sys*. Linux file managers have settings to indicate whether or not to display hidden files. Samba also has a configuration option to indicate whether or not Linux's hidden files should be seen as hidden files by Windows.